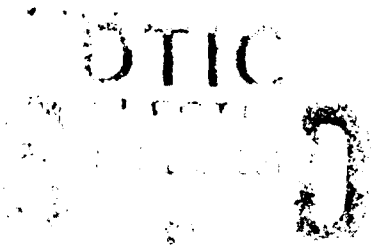


Appearance Simulator for Computer Vision Research

Yoshimasa Fujiwara, Shree Nayer, and Katsushi Ikeuchi

CMU-RI-TR-91-16

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

August 1991

This research was conducted in the Task-oriented Vision Laboratory, Vision and Autonomous Center, the Robotics Institute, Carnegie Mellon University. Image Understanding Research in the TVL is supported in part by the Defense Advanced Research Project Agency, DOD, through ARPA order No.4976, and monitored by the Air Force Avionics Laboratory under contract F33615-87-C-1499.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	August 1991	technical

4. TITLE AND SUBTITLE

Appearance Simulator for Computer Vision Research

5. FUNDING NUMBERS

F33615-87-C-1499
No. 4976

6. AUTHOR(S)

Yoshimasa Fujiwara, Shree Nayer, and Katsushi Ikeuchi

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

8. PERFORMING ORGANIZATION REPORT NUMBER

CMU-RI-TR-91-16

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Avionics Laboratory
DARPA

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release;
Distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

Computer vision research needs a flexible and robust method to obtain accurate appearance of a scene. Such needs include generating test data, using a simulator as a part of an algorithm, and generating model data for object recognition and inspection. A sophisticated and reasonable mechanism for generating scene appearance is a technique known as ray tracing. However, current ray tracers are mainly developed for computer graphic research. They are based on ad hoc reflectance mechanisms; their algorithms emphasize computational costs by choosing computational expediency at the expense of an appearance grounded in physics.

We are developing an appearance simulator to be able to produce a scene according to the laws of physics. By combining the reflectance theory based on physics and the technique of ray tracing, we are able to produce scene appearances that are adequate and realistic enough for computer vision. First, we will briefly describe our reflectance model to be used in our simulator. Then, we will discuss several implementation issues how to embody the model into real algorithms. Finally, we will generate several scene appearance to verify the performance of our system and illustrate applications of this system.

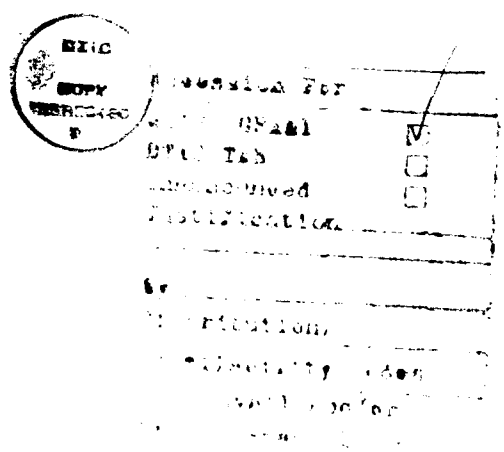
14. SUBJECT TERMS

15. NUMBER OF PAGES
30 pp

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
unlimited	unlimited	unlimited	unlimited

Contents



A-1

List of figures

Abstract

Computer vision research needs a flexible and robust method to obtain accurate appearance of a scene. Such needs include generating test data, using a simulator as a part of an algorithm, and generating model data for object recognition and inspection. A sophisticated and reasonable mechanism for generating scene appearance is a technique known as ray tracing. However, current ray tracers are mainly develop for computer graphic research. They are based on ad hoc reflectance mechanisms; their algorithms emphasize computational costs by choosing computational expediency at the expense of an appearance grounded in physics.

We are developing an appearance simulator to be able to produce a scene according to the laws of physics. By combining the reflectance theory based on physics and the technique of ray tracing, we are able to produce scene appearances that are adequate and realistic enough for computer vision. First, we will briefly describes our reflectance model to be used in our simulator. Then, we will discuss several implementation issues how to embody the model into real algorithms. Finally, we will generate several scene appearance to verify the performance of our system and illustrate applications of this system.

1 Introduction

Computer vision research needs a flexible and robust method to obtain accurate appearances of a scene. Such needs include:

- *generating test data* - verification of a newly developed algorithm needs several appearance data to which we apply the algorithm. Although it is possible to obtain real data under a highly controlled environment, such methods often give us difficulties to choose several photometric, geometric, and noise parameters.
- *using a simulator as a part of an iterative recovering algorithm* - an iterative algorithm often requires a component to generate appearances based on a set of parameters determined by the algorithm and then adjust the parameters from the difference between generated and obtain appearances iteratively.
- *generating model data* - a model based object recognition and inspection algorithm often need model appearances from which we can extract necessary features to be used for recognition and inspection.

A sophisticated and reasonable mechanism for generating scene appearances is a technique known as ray tracing [1, 4, 10]. Ray tracing method employs the principles of geometric optics. For each pixel on the screen, a ray of infinitesimal width is traced from the focal point through the pixel, into object space. The ray is tested against each object in the database, and the object for which this intersection is the closest to the focal point is used to calculate the intensity of the pixel on the screen. Whenever it is determined that a ray intersects a surface, a secondary ray is first traced from the intersection point toward each light sources. If the ray encounters an object between surface and light source, the intersection point can be considered to be in shadow and the intensity of the pixel diminished according to the illumination model. A ray can also be traced in the direction of reflection. If the reflected ray intersects a surface, the closest surface intersected by this ray is determined , and a portion of the illumination from this second surface is then used in determining the total intensity of the pixel. This adds a level of recursion to the algorithm in that a reflected ray generates additional shadow rays and possible other reflected rays.

Current ray tracers are mainly develop for computer graphic research. They are based on ad hoc reflectance mechanisms such as Phong's model [9]; their algorithms emphasize computational costs by choosing computational expediencey at the expense of an appearance grounded in physics. The end result is that ray tracing does not realistically enough, for our purposes, model appearances of scene. The goal of ray tracing is to produce a scene which looks natural and realistic to *human beings*. While such a goal is adequate enough for human perception and recognition, it is not accurate enough to serve as a simulator for computer vision projects. Vision research needs to have the capability of simulation that goes beyond creating scenes looking natural to human beings.

This paper will develop an appearance simulator to be able to produce a scene according to the laws of physics. By combining the reflectance theory based on physics and the technique of ray tracing, we are able to produce scene appearances that are adequate and realistic enough for computer vision.

The remainder of this paper consists of three sections. The second section briefly describes our reflectance model to be used in our simulator. The third section handles several implementation issues how to embody the model into real algorithms. Finally, we will generate several scene appearances to verify the performance of our system.

2 Reflection Model

Currently, there are two main approaches to model reflectance mechanism: physical and geometrical optics. Physical optics is based directly on electromagnetic wave theory and uses Maxwell's equations to study the propagation of light. In particular, the Beckmann-Spizzichino reflectance model solves Maxwell's equations using the Helmholz integral by assuming a perfect conductor surface [3]. On the other hand, geometric optics, such as the Torrance-Sparrow [11], analyze a path of light ray to approximate the specular component of reflection for rough surfaces. These the Beckmann-Spizzichino physical optics model and the Torrance-Sparrow geometric optics model have been found to fit experimental data quite well [6, 11].

Wehave proposed an unified reflectance model based on the Beckmann-Spizzichino and Torrance-Sparrow models that also contains the Lambertian mechanism [8]. This model comprises three reflection components: the diffuse lobe, the specular lobe and the specular spike. We report identical results for both Beckmann-Spizzichino and Torrance-Sparrow models under certain restrictions, and conclude that, whenever the Torrance-Sparrow is simpler to formulate, the Beckmann- Spizzichino model is more general and explains better all the reflection mechanisms for different kind of surfaces, from smoother to rougher. This paper will use this unified model to get better image of rough surface and intermediate surface.

2.1 Beckmann-Spizzichino specular reflection model

Beckmann and Spizzichino only use the mean field $< E_2 >$ as a stepping stone to derive the mean scattered power $< E_2 E_2^* > = < \mid E_2 \mid^2 >$. For an angle on incidence θ_i, the mean power scattered in the direction (θ_r, ϕ_r) by a rough surface, whose height h is normally distributed with mean value $< h > = 0$, standard deviation σ_h, and correlation distance T, is given by

$$< E_2 E_2^* > = \frac{E_o^2 A^2 \cos^2\theta_i}{\lambda^2 R_o^2} e^{-g} \left(\rho_o^2 + \frac{\pi T^2 D^2}{A} \sum_{m=1}^{m=\infty} \frac{g^m}{m!m} e^{-v_{xy}^2 T^2/4m} \right) \quad (1)$$

where

$$g = \left(2\pi \frac{\sigma_h}{\lambda} (\cos\theta_i + \cos\theta_r) \right)^2 \quad (2)$$

$$\rho_o = sinc\,(v_x X)\; sinc\,(v_y Y) \quad (3)$$

$$D = \left(\frac{1 + \cos\theta_i \cos\theta_r - \sin\theta_i \sin\theta_r \cos\phi_r}{\cos\theta_i (\cos\theta_i + \cos\theta_r)} \right) \quad (4)$$

$$v_{xy} = \sqrt{v_x^2 + v_y^2} \quad (5)$$

We see from equation (2) that the factor g in equation (1) depends on λ. Raleigh criterion states that the roughness of a surface can be expressed as a function of λ. In this sense, g represents the roughness of the surface and the three cases $g \ll 1$, $g \approx 1$, and $g \gg 1$ correspond to smooth[1] surfaces, moderately rough surfaces, and rough surfaces, respectively. It is important to note that the model under consideration only attempts to describe the reflection mechanism that is often referred to by the vision research community as "specular reflection". As seen from equation (1), the mean scattered power is the sum of two terms. The first term $e^{-g}\rho_o^2$ is the "specular spike" component of the specular reflection and represents the only the field scattered very close to the specular direction. It is seen from equation (3), that when the surface dimensions are small, ρ_o becomes a very sharp function of θ_i and $\theta_r$and is equal to zero for any direction of scattering except a narrow range about the specular direction. Since the mean slope of the surface is constant and is independent of the roughness of the surface, a privileged scattering in the specular direction is expected. The second term in equation (1) corresponds to the "specular lobe"[2], i.e. the diffusely scattered field that results from the roughness of the surface. The specular lobe component is distributed around the specular direction. For a perfectly smooth surface, $g = 0$ and the specular lobe vanishes while the specular spike is strong. As the roughness measure g increases, the spike component shrinks rapidly, while the lobe component increases in magnitude. The exponential series given by the summation in the lobe component may be approximated for smooth ($g \ll 1$) and very rough ($g \gg 1$) surfaces. The approximations result in simpler expressions for the scattered power for these two extreme surface conditions:

$$< E_2 E_2^* >_{smooth} = \frac{E_o^2 A^2 \cos^2\theta_i}{\lambda^2 R_o^2} e^{-g} \left(\rho_o^2 + \frac{\pi T^2 D^2 g}{A} e^{-v_{xy}^2 T^2/4} \right) \quad (g \ll 1) \quad (6)$$

$$< E_2 E_2^* >_{rough} = \frac{E_o^2 A \cos^2\theta_i \pi T^2 D^2}{\lambda^2 R_o^2 v_z^2 \sigma_h^2} exp\left(\frac{-v_{xy}^2 T^2}{4 v_z^2 \sigma_h^2} \right) \quad (g \gg 1) \quad (7)$$

The above equations for scattered power represent the Beckmann-Spizzichino reflectance model. Anyway, we are interested in the radiance of the surface, L_r which is defined as the energy of light emitted per unit fore-shortened area per unit solid angle. Surface radiance can be related to image irradiance which we are specially interested in, because we want to simulate an image taking from a given scene. Image irradiance is defined as the incident flux density in a surface of the image plane.

Consider the image formation geometry shown in Figure 1. The surface radiance

[1] We define a smooth surface as one that is either perfectly smooth or "slightly" rough.

[2] Beckmann and Spizzichino have referred to this component as the "diffuse" component. The term "diffuse" has historically been used by the vision community to describe the reflection component that results from other mechanisms such as multiple reflections and internal scattering. To avoid confusion we will refer to the diffuse component of specular reflection as the specular lobe.

derived for this geometry [5] is given by the following expression:

$$L_r = \frac{1}{2}\sqrt{\frac{\mu}{\varepsilon}}\frac{R_o^2 f^2 < E_2 E_2^* >}{z^2 dA_i \cos\gamma}. \tag{8}$$

where ε and μ are the *permittivity* and *permeability* of the medium in which the wave is propagated, and dA_i represents the image element area (pixel) which is constant for a given sensor.

By substituting equation (1) in (8) we obtain:

$$L_r = \sqrt{\frac{\mu}{\varepsilon}}\frac{E_o^2 \cos^2\theta_i}{2\lambda^2 \cos\gamma} e^{-g}\left(\left(\frac{z}{f}\right)^2 \frac{dA_i}{\cos^2\theta_r}\rho_o^2 + \frac{\pi T^2 D^2}{\cos\theta_r}\sum_{m=1}^{m=\infty}\frac{g^m}{m!m} e^{-v_{xy}^2 T^2/4m}\right) \tag{9}$$

Similarly, from equations (6) and (7), the surface radiance for smooth and rough surfaces may be written as

$$L_{rsmooth} = \sqrt{\frac{\mu}{\varepsilon}}\frac{E_o^2 \cos^2\theta_i}{2\lambda^2 \cos\gamma} e^{-g}\left(\left(\frac{z}{f}\right)^2 \frac{dA_i}{\cos^2\theta_i}\rho_o^2 + \frac{\pi T^2 D^2 g}{\cos\theta_r}\right) \quad (g \ll 1) \tag{10}$$

$$L_{rrough} = \sqrt{\frac{\mu}{\varepsilon}}\frac{E_o^2 \cos^2\theta_i \pi T^2 D^2}{2\lambda^2 \cos\gamma \cos\theta_r v_z^2 \sigma_h^2} exp\left(\frac{-v_{xy}^2 T^2}{4 v_z^2 \sigma_h^2}\right) \quad (g \gg 1) \tag{11}$$

We can also obtain $f_r(\theta_i; \theta_r, \phi_r)$, the BRDF (bidirectional reflectance distribution function), of the surface using its radiance L_r and irradiance I_s as $f_r = L_r/I_s$. We see that surface irradiance I_s is defined as the light energy incident per unit area of the surface. The surface irradiance can be obtained as

$$I_s = S_a \cos\theta_i = \frac{1}{2}\sqrt{\frac{\mu}{\varepsilon}} < E_1 E_1^* > \cos\theta_i \tag{12}$$

where the term $\cos\theta_i$ accounts for the fact that the same amount of incident energy is received by a greater surface area when the angle of incidence θ_i is increased and E_1 is the scalar value of the incident plane wave.

The BRDF of rough surface $f_{r_{SL}}$ is determined with equations (11) and (12).

$$f_{r_{SL}}(\theta_i, \phi_i, \theta_r, \phi_r) = \frac{\cos\theta_i \pi T^2 D^2}{\lambda^2 \cos\theta_r v_z^2 \sigma_h^2} exp\left(\frac{-v_{xy}^2 T^2}{4 v_z^2 \sigma_h^2}\right) \tag{13}$$

The BRDF of smooth surface $f_{r_{SL}}$ is determined with equation (10) and (12). Houchence uses delta function instead of sinc function used in equation (3) and obtains BRDF of metallic surface in [6].

$$f_{r_{ss}}(\theta_i, \phi_i, \theta_r, \phi_r) = \frac{exp(-(4\pi\frac{\sigma}{\lambda}cos\,\theta_i)^2)}{cos\,\theta_i d\omega_i} U(\theta_r - \theta_i)\, U(\theta_i + d\theta_i - \theta_r)$$
$$U(\phi_r - (\phi_i + \pi))\, U((\phi_i + d\phi_i + \pi) - \phi_r) \qquad (14)$$

$$where \quad U(z) = \begin{cases} 0, & z < 0 \\ 1, & z \geq 0 \end{cases}$$

Using the same imaging geometry shown in Figure 1, Horn [5] has established a relationship between surface radiance L_r and image irradiance I_{im}. The image irradiance is found to be proportional to surface radiance and can be expressed as

$$I_{im} = L_r \frac{\pi}{4} \left(\frac{d}{f}\right)^2 cos^4\gamma\,. \qquad (15)$$

When the image covers only a narrow angle of the scene, we see that $\gamma \approx 0$ and it is reasonable to assume that $cos\gamma = 1$ in the above equations.

2.2 Lambertian diffuse reflection model

When light strikes on a surface area dA_s like in Figure 2, some of the light passes through the surface and the remaining portion is reflected. Light penetrating the surface area hits internal pigments of an object, is partly absorbed, and is randomly emitted. We will refer to this reflection component as the *Lambertian component*. The incident flux on the surface, $d^2\Phi_i$ is given as $L_i d\omega_i dA_s cos\theta_i$. The surface irradiance, I_s is determined by dividing the incident flux by the surface area, dA_s . Thus,

$$I_s = L_i d\omega_i cos\theta_i.$$

Since L_r , the surface radiance given by the Lambertian component, is proportional to the irradiance of the surface given by a light source [7], and K_{lamb} is a proportional constant, we can state the following formula:

$$L_{r,lamb} = K_{lamb} L_i d\omega_i cos\theta_i \qquad (16)$$

This equation is known as Lambertian's cosine law.

As shown in [5] , BRDF of Lambertian reflection is known as $1/\pi$.

$$f_{r_{Lm}}(\theta_i, \phi_i, \theta_r, \phi_r) = \frac{1}{\pi} \qquad (17)$$

3 Radiometric Brightness Calculation in Ray Tracing

This section describes the complete ray tracing algorithm base on the reflectance models described in the previous section. To give physical meaning to the ray tracer, we use the radiometric calculation, considering in-coming and out-going light energy at each surface patch, and get image brightness based on in-coming energy of light. In particular, this section will consider how to implement the reflectance mechanism, how to calculate brightness radiometrically, and how to handle a point light source in the specular spike model.

3.1 Basic radiometry calculation - one surface reflection

Radiometrically radiance towards (θ_r, ϕ_r) is denoted as following,

$$L_r(\theta_r, \phi_r) = \iint (K_{Lm} f_{r_{Lm}}(\theta_i, \phi_i, \theta_r, \phi_r) + K_{SS} f_{r_{SS}}(\theta_i, \phi_i, \theta_r, \phi_r) + K_{SL} f_{r_{SL}}(\theta_i, \phi_i, \theta_r, \phi_r)) I_s(\theta_i, \phi_i)\, d\omega_i \quad (18)$$

where K_{Lm} , K_{SS} , K_{SL} are albedo for Lambertian, specular spike and specular lobe, respectively.

As shown in Figure 3 , $f_{r_{SS}}$ and $f_{r_{SL}}$ are distributed along the specular direction. $f_{r_{SS}}$ effective area is very sharp and $f_{r_{SL}}$ effective area is wider. $f_{r_{Lm}}$ is same in all directions.

First we show simplest example which made of one surface and one light source. Then we show multiple light sources case. We also show another example which has two surfaces and one light source. After that we show how we realize physical reflection models in ray tracing.

Consider the case shown in Figure 4. There is just one light source and one surface. A ray tracing vector $\mathbf{R}_f$ which starts from focal point F and goes through center of one image pixel intersects with the surface at point P_s . The radiance of P_s towards focal point F , direction (θ_r, ϕ_r) , is denoted as $L_{r1}(\theta_r, \phi_r)$ and gives intensity of the pixel. In this case the radiance at P_s is caused only by the light source L_1 .
As we assume the light is a point source,

$$L_{r1}(\theta_r, \phi_r) = (f_{r_{Lm}}(\theta_{i1}, \phi_{i1}, \theta_r, \phi_r) + f_{r_{SS}}(\theta_{i1}, \phi_{i1}, \theta_r, \phi_r) + f_{r_{SL}}(\theta_{i1}, \phi_{i1}, \theta_r, \phi_r)) I_s(\theta_{i1}, \phi_{i1}) \quad (19)$$

where (θ_{i1}, ϕ_{i1}) is the direction of L_1 measured from P_s , $I_s(\theta_{i1}, \phi_{i1})$ is irradiance at P_s given by the light source. $I_s(\theta_{i1}, \phi_{i1})$ is

$$I_s(\theta_{i1}, \phi_{i1}) = L_{r_{LIGHT1}}\, d\omega_i \cos\theta_{i1} \quad (20)$$

$$\text{where} \quad d\omega_i = \frac{\pi r^2}{d^2}$$

$L_{r_{LIGHT}}$ is radiance of the light source, $d\omega_i$ is spatial angle of the light source from P_s , r is radius of the light source, d is distance between P_s and the light source.

If there are N light sources, radiance is

$$L_{r1}(\theta_r, \phi_r) = \sum_{k=1}^{N}((f_{r_{Lm}}(\theta_i^k, \phi_i^k, \theta_r, \phi_r) + f_{r_{SS}}(\theta_i^k, \phi_i^k, \theta_r, \phi_r) + f_{r_{SL}}(\theta_i^k, \phi_i^k, \theta_r, \phi_r))I_s(\theta_i^k, \phi_i^k)) \quad (21)$$

where (θ_i^k, ϕ_i^k) is the direction of k-th light source from P_s , $I_s(\theta_i^k, \phi_i^k)$ is irradiance at P_s from k-th light source. $I_s(\theta_i^k, \phi_i^k)$ is

$$I_s(\theta_i^k, \phi_i^k) = L_{r_{LIGHT}}^k d\omega_i^k \cos\theta_i^k \quad (22)$$

$$where \quad d\omega_i^k = \frac{\pi r_k^2}{d_k^2}$$

$L_{r_{LIGHT}}^k$ is radiance of k-th light source, $d\omega_i^k$ is spatial angle of k-th light source from P_s , r_k and d_k are radius and distance from P_s of k-th light source, respectively.

3.2 Basic radiometry calculation - multiple surface reflections

Consider the ray tracing geometry shown in Figure 5. There is another surface which can reflect light effectively toward point P_s . So radiance at P_s towards focal point F is

$$L_r(\theta_r, \phi_r) = L_{r1}(\theta_r, \phi_r) + L_{r2}(\theta_r, \phi_r) \quad (23)$$

where L_{r1} and L_{r2} are radiance caused by the light source and interreflection, respectively. To get L_{r2} , we have to calculate the (18) because next surface is too big to be considered as a point. It means we have to care about all the light which incident to a point, irradiance at that point, to get radiance of the point, theoretically. As can be seen in Figure 5, each BRDF has its effective area, i.e. out of such area, angle, BRDF is negligible. We modify (18) based on this concept, BRDF effective area.

$$\begin{aligned} L_{r2}(\theta_r, \phi_r) &= K_{SS} \int\int_{SS} f_{r_{SS}}(\theta_i, \phi_i, \theta_r, \phi_r) I_s(\theta_i, \phi_i) d\omega_i \\ &+ K_{SL} \int\int_{SL} f_{r_{SL}}(\theta_i, \phi_i, \theta_r, \phi_r) I_s(\theta_i, \phi_i) d\omega_i \\ &+ K_{Lm} \int\int_{Lm} f_{r_{Lm}}(\theta_i, \phi_i, \theta_r, \phi_r) I_s(\theta_i, \phi_i) d\omega_i \end{aligned} \quad (24)$$

SS, SL, Lm denote effective area of specular spike, specular lobe, Lambertian, respectively. In another words, we calculate each component separately within their own effective area using ray tracing to find surface data such as K_{SS}, K_{SL}, K_{Lm} and geometric data such as θ_r, ϕ_r, θ_i,

ϕ_i. Also we can distinguish light sources from surfaces and omit light sources from above calculation of L_{r2} . Irradiance $I_s(\theta_i, \phi_i)$, irradiance at P_s comes from (θ_i, ϕ_i) direction, can be calculated with radiance from P_s' , the intersection point of next surface and vector from point P_s toward direction (θ_i, ϕ_i) . That is

$$I_s(\theta_i, \phi_i) = L_r'(\theta_r', \phi_r')\, d\omega \cos\theta_i \tag{25}$$

where $L_r'(\theta_r', \phi_r')$ is radiance at P_s' which is on the next surface toward P_s , direction (θ_r', ϕ_r') . As there is no another surface which can reflect light toward P_s' effectively, we just have to care about the light from light source. So $L_r'(\theta_r', \phi_r')$ is

$$\begin{aligned} L_r'(\theta_r', \phi_r') &= L_{r1}'(\theta_r', \phi_r') && (26) \\ &= (K_{SS}' f_{r_{SS}}'(\theta_{i1}', \phi_{i1}', \theta_r', \phi_r') + (K_{SL}' f_{r_{SL}}'(\theta_{i1}', \phi_{i1}', \theta_r', \phi_r') \\ &\quad + (K_{Lm}' f_{r_{Lm}}'(\theta_{i1}', \phi_{i1}', \theta_r', \phi_r')) I_s'(\theta_{i1}', \phi_{i1}') && (27) \end{aligned}$$

where L_{r1}' is radiance at P_s' caused by the light source, K_{Lm}' , K_{SS}' , K_{SL}' are albedo at P_s' , $f_{r_{Lm}}'$, $f_{r_{SS}}'$, $f_{r_{SL}}'$ are BRDF at P_s' , $(\theta_{i1}', \phi_{i1}')$ is the direction toward the light source from P_s' . $I_s'(\theta_{i1}', \phi_{i1}')$ is irradiance at P_s' caused by the light source.

$$I_s'(\theta_{i1}', \phi_{i1}') = L_{r_{LIGHT}}\, d\omega_i' \cos\theta_{i1}' \tag{28}$$

$d\omega_i'$ is spatial angle of the light source seen from P_s' .

If there are yet another surface which we should consider as the next surface, $L_r'(\theta_r', \phi_r')$ becomes

$$L_r'(\theta_r', \phi_r') = L_{r1}'(\theta_r', \phi_r') + L_{r2}'(\theta_r', \phi_r') \tag{29}$$

where L_{r2}' is radiance at P_s' caused by interreflection which is

$$\begin{aligned} L_{r2}'(\theta_r, \phi_r) &= K_{SS}' \int\int_{SS} f_{r_{SS}}(\theta_i', \phi_i', \theta_r', \phi_r') I_s'(\theta_i', \phi_i')\, d\omega_i \\ &+ K_{SL}' \int\int_{SL} f_{r_{SL}}(\theta_i', \phi_i', \theta_r', \phi_r') I_s'(\theta_i', \phi_i')\, d\omega_i \\ &+ K_{Lm}' \int\int_{Lm} f_{r_{Lm}}(\theta_i', \phi_i', \theta_r', \phi_r') I_s'(\theta_i', \phi_i')\, d\omega_i \end{aligned} \tag{30}$$

where $I_s'(\theta_i', \phi_i')$, irradiance at P_s' is

$$I_s'(\theta_i', \phi_i') = L_r''(\theta_r'', \phi_r'')\, d\omega \cos\theta_i' \tag{31}$$

$L_r''(\theta_r'', \phi_r'')$ is radiance at P_s'' , intersection point of the third surface and vector starts from P_S' toward (θ_i', ϕ_i') direction.
This process for $L_r'(\theta_r', \phi_r')$, from equation (29) to (31), is completely same as that for $L_r(\theta_r, \phi_r)$,

from equation (23) to (25). This iteration will be continued until there is no more next surface which can reflect light effectively to the point.
We have briefly explained how we radiometricly calculate the image, i.e. radiance at the each pixel, using BRDF equations (17), (14), (13) based on radiometric theories in ray tracing.
Next subsections show how we realize those BRDF equations, specular spike, specular lobe, Lambertian.

3.3 Calculating specular spike BRDF

The effective area of specular spike BRDF is very narrow and on the reflection direction $(\theta_r, \phi_r + \pi)$. We assume it's narrow enough to be considered as a single ray. So we only search along the reflection direction, i.e. make the secondary ray tracing vector from P_s toward this direction to find the intersection point P_s' on next surface. The first portion of right handed side of equation(24) can be obtained as

$$K_{SS} \int\int_{SS} f_{rss}(\theta_i, \phi_i, \theta_r, \phi_r) I_s(\theta_i, \phi_i) \, d\omega_i = K_{SS} f_{rss}(\theta_r, \phi_r + \pi, \theta_r, \phi_r) I_s(\theta_r, \phi_r + \pi) \tag{32}$$

$$\textit{where} \quad I_s(\theta_r, \phi_r + \pi) = L_r'(\theta_r', \phi_r') \, d\omega \cos\theta_r \tag{33}$$

where $L_r'(\theta_r', \phi_r')$ is radiance of P_s' toward direction (θ_r', ϕ_r') , i.e. the direction from P_s' toward P_s , $d\omega$ is the spatial angle which is set as 0.087, it is the angle of 5° cone, according to the effective area of specular spike BRDF.
By substituting equation (14) and (33) to (32), we get following equation for specular spike component of radiance at P_s toward the direction (θ_r, ϕ_r) .

$$K_{SS} \int\int_{SS} f_{rss}(\theta_i, \phi_i, \theta_r, \phi_r) I_s(\theta_i, \phi_i) \, d\omega_i = K_{SS} \frac{exp(-(4\pi \frac{\sigma}{\lambda} cos\,\theta_r)^2)}{cos\,\theta_r \, dwi} U(\theta_r - \theta_r) \, U(\theta_r + d\theta_i - \theta_r)$$
$$U(\phi_r - ((\phi_r + \pi) + \pi)) \, U((((\phi_r + \pi) + d\phi_i + \pi) - \phi_r) \, L_r'(\theta_r', \phi_r') \, d\omega \cos\theta_r$$
$$= K_{SS} \, exp(-(4\pi \frac{\sigma}{\lambda} cos\,\theta_r)^2) \, L_r'(\theta_r', \phi_r') \tag{34}$$

If the surface is very smooth, σ becomes almost 0 and equation (34) becomes

$$K_{SS} \int\int_{SS} f_{rss}(\theta_i, \phi_i, \theta_r, \phi_r) I_s(\theta_i, \phi_i) \, d\omega_i = K_{SS} L_r'(\theta_r', \phi_r') \tag{35}$$

This means the surface reflects same amount of energy which comes from next surface point P_s' exactly like a miller.

As seen in Figure 6, ray tracing usually calculates the intensity at the center of pixels using only one vector which goes though the center of each pixel assuming that objects are big enough to cover entire single pixel or light from the object becomes negligibly weak when object is far away and seen too small to cover the pixel. But for light sources, it's not so as

they are very bright and small. Pixels should become very bright if light source is included in sight of the pixels but with ray tracing light source may be out of the narrow effective area of specular spike BRDF which is laid aling the ray tracing vector if it is not close enough to the center of the pixel even though image of a light source is actually in a pixel. As can be seen in Figure 7, we add 4 more vectors to each ray tracing vector to check if light sources are included in the sight of each pixel. Those, denoted with fine break lines in Figure 7, are generated from focal point toward corner points of each pixel and reflected same as ray tracing vector. If a light source is found within those 4 vectors, irradiance is calculated as if the light source is seen at the center of the pixel so that the pixel becomes bright. It is reasonable because some point on the surface which is inside of the sight of a pixel should have reflection direction directly toward the light source, like vector **X** in Figure 7, if the light source is among the 4 corner vectors.

3.4 Calculating specular lobe BRDF

The effective area of specular lobe BRDF is wider than which of specular spike. So we use some number of ray tracing vectors distributed evenly in the effective area instead of a single vector so that we can realize the integration of irradiances from various directions. Then we translate those irradiances to one particular radiance from P_s toward (θ_r, ϕ_r) direction.

Now the second portion of right handed side of equation(24) is

$$K_{SL} \int\int_{SL} f_{rSL}(\theta_i, \phi_i, \theta_r, \phi_r) I_s(\theta_i, \phi_i) \, d\omega_i = K_{SL} \sum_{n}^{N} (f_{rSL}(\theta_i^n, \phi_i^n, \theta_r, \phi_r) I_s(\theta_i^n, \phi_i^n)) \tag{36}$$

where (θ_i^n, ϕ_i^n) denotes the direction of n–th ray tracing vector from P_s.

$I_s(\theta_i^n, \phi_i^n)$ is

$$I_s(\theta_i^n, \phi_i^n) = L_r'(\theta_r', \phi_r') \, d\omega \cos\theta_i^n \, U(\frac{\pi}{2} - \theta_i^n) \tag{37}$$

By substituting equation (13) and (37) to (36), we get following equation for specular lobe component of radiance toward the direction (θ_r, ϕ_r) .

$$K_{SL} \int\int_{SL} f_{rSL}(\theta_i, \phi_i, \theta_r, \phi_r) I_s(\theta_i, \phi_i) \, d\omega_i =$$

$$K_{SL} \sum_{n}^{N} \left(\frac{\cos\theta_i^n \, \pi T^2 D^2}{\lambda^2 \cos\theta_r \, {v_z}^2 \, {\sigma_h}^2} \exp\left(\frac{-{v_{xy}}^2 T^2}{4{v_z}^2 {\sigma_h}^2}\right) L_r'(\theta_r', \phi_r') \, d\omega \cos\theta_i^n \, U(\frac{\pi}{2} - \theta_i^n)\right) \tag{38}$$

To make such distributed ray tracing vectors, we make a series of vectors $\mathbf{C}_i$ around Z axis and swing them along the reflection direction.

$$\mathbf{C}_i = \begin{cases} [0, 0, 1]^T & ; i = 0 \\ [\sin(m\,\delta\theta)\cos\frac{i\delta\phi}{m}, \sin(m\,\delta\theta)\sin\frac{i\delta\phi}{m}, \cos(m\,\delta\theta)]^T & ; i \neq 0 \end{cases} \tag{39}$$

$$\begin{cases} m = [\frac{(i-1)}{6}] + 1 \\ j = i - 1 - 6(m-1) \\ \delta\theta = 5^\circ \\ \delta\phi = 60^\circ \end{cases}$$

Vectors for actual search $\mathbf{C}_i'$ should be distributed around the reflection direction $\mathbf{R}$. Vector $\mathbf{R}$ is

$$\mathbf{R} = \frac{\frac{1}{(\mathbf{V}_{rt} \cdot \mathbf{N})} \mathbf{V}_{rt} + 2\mathbf{N}}{\left| \frac{1}{(\mathbf{V}_{rt} \cdot \mathbf{N})} \mathbf{V}_{rt} + 2\mathbf{N} \right|} \tag{40}$$

where $\mathbf{V}_{rt}$ is unit ray tracing vector comes to the point P_s from previous surface, $\mathbf{N}$ is the unit normal vector at that point.

By the formula of Rodorigues, we can obtain $\mathbf{C}_i'$.

$$\mathbf{C}_i' = cos\,\psi\,\mathbf{C}_i + sin\,\psi\,(\omega \times \mathbf{C}_i) + (1 - cos\,\psi)\,(\omega \cdot \mathbf{C}_i)\,\omega \tag{41}$$

$$\begin{cases} \psi = cos^{-1}(\mathbf{Z} \cdot \mathbf{R}) \\ \omega = (\mathbf{Z} \times \mathbf{R}) \end{cases}$$

We also need angles θ_i^n and ϕ_i^n for the calculation of (38). Those are

$$\theta_i^n = cos^{-1}(\mathbf{C}_n' \cdot \mathbf{N}) \tag{42}$$

$$\phi_i^n = cos^{-1}\left(\frac{(\mathbf{C}_n' \times \mathbf{N})}{|\,(\mathbf{C}_n' \times \mathbf{N})\,|} \cdot \frac{(\mathbf{V}_{rt} \times \mathbf{N})}{|\,(\mathbf{V}_{rt} \times \mathbf{N})\,|}\right) + \phi_r \tag{43}$$

3.5 Calculating Lambertian BRDF

As our attention is mainly on the specular reflection, we don't calculate diffuse or Lambertian component very precisely. We just calculate Lambertian component caused by light sources assuming that is the dominant source of irradiance.

So the third portion of right handed side of equation(24) is

$$K_{Lm} \int\int_{Lm} f_{r_{LM}}(\theta_i, \phi_i, \theta_r, \phi_r)\, I_s(\theta_i, \phi_i)\, d\omega_i \simeq K_{Lm} \frac{1}{\pi} \int\int_{Lm} \emptyset\, d\omega_i = 0 \tag{44}$$

4 Experiments

We can use our appearance simulator for several different applications. The first application is to generate a physically based appearance of a scene based on surface roughness parameters. The second application is to generate a necessary information for model based vision such as a synthesized depth map or a map indicating interreflection relation. The third application is to examine a real range data by generating a sequence of simulated images for animation as a part of our rapid prototyping project.

4.1 Simulating Appearances

Light source image on rough surface We examine the performance of our system to handle appearances of intermediate rough surfaces. We created an image by our appearance simulator using the following parameters: correlation distance of surface roughness, T is 3.0^{-6} m and standard deviation of surface σ is 1.0^{-7} m, and light wave length λ is 5.55^{-6} m . The created image as shown in Figure 8 has two kinds of specular reflection, specular lobe and spike component, at the same time.

Figure 9 shows an actual image of a bowl whose surface is determined as intermediate rough. In the dull specular area corresponding to the specular lobe reflection, we can observe a small bright spot corresponding to the specular spike reflection of light source image as predicted by our simulator. Usual ray tracers cannot simulate those two kinds of specular reflections.

Light source image on smooth surface By changing the parameters, we can generate a well-known spot specular reflection by using the same appearance simulator. Figure 10 shows the created image of smooth sphere. There is only bright spot which is actually the reflected image of light source. Parameter T is 1.0^{-8} m and σ is 1.0^{-9} m.

Interreflection between rough surfaces We demonstrate the ability of our system to handle appearances of interreflection between intermediate rough surfaces. Figure 11 is an image created by our simulator. The interreflection reflection between intermediately rough surface can be seen. Parameter T is 1.0^{-8} m and σ is 1.0^{-9} m.

The similar interreflection can be observed in Figure 12 of an actual image of intermediatly rough sphere on intermediately rough plane.

4.2 Synthesizing depth maps

We can generate a depth map by using our appearance simulator from a representation given by our geometric modeler Vantage [2].

Figure 13 shows the line drawing given by Vantage. From this representation, the system generates a scene appearance as shown in Figure 14 and a depth map of the scene as shown in Figure 15.

Each plana. and curved surface has an unique identification number (ID number) in our simulator. The sy stem generates a map of surface id numbers at each pixel; which surface is visible at each pixel position. Figure 16 shows such a distribution of primal face ID number. This map can bc regarded as a perfect segmentation of the scene into each planar or curved surface.

The system also generates a map of surface numbers to which surface each ray hits the second time, after reflected by the first surface. Figure 17 shows such a distribution of secondary face ID number. By comparing these two distributions, we can establish the interreflection relationship beiween surfaces at eacli pixel.

4.3 Examining range data

We are developing a system to generate a bust from a range data. As shown in Figure 18, the system consists of two parts: data acquistion and bust generation. The range data is obtained by our range finger. Then, the data is examined whether it is appropriate for later bust generation process by using our system. The data is converted into a certain format and given to the stereolithography system by which we can make a bust of the human.

For this system, we prepare an interface routine between the range data acquisition system and our ray tracer system. We obtain the data of a human by using the range finder. Figure 19(a) shows a brightness image of a human face, while Figure 19(b) shows the isoplot image of the depth distribution.

From the depth map of the human, we can generate a simulated image of the human. Figure 20(a) shows the simulated plaster image, while Figure 20(b) shows the simulated stainless steel image. This capability is used to examine whether the obtained data is accurate enough for making a bust of the human using the stereolithography system.

Figure 21 shows the distance distribution from an arbitrary focal point. Note that the new focal point is different from the original focal point.

5 Conclusion

We have developed an appearance simulator based on a physically based reflectance model. By using this appearance simulator, we can generate

- physically based appearances for vision algorithms,
- depth and interreflection map from a scene representation given by a geometric modeler Vantage, and
- animation images used to examine data consistency for making a bust from a range data.

Our future directions include to increase the efficiency of computation and to accommodate more accurate camera characteristics such as defocus effect and lens distortion.

Acknowledgement

Takeo Kanade provided useful comments. Kathryn Porsche proofread drafts of this paper and provided many useful comments which have improved the readability of this paper. The authors also thank Tetsuo Kiuchi, Ye Mei, and Carlos Cerrada for their valuable comments.

This research was conducted in the Task-oriented Vision Laboratory, Vision and Autonomous Center, the Robotics Institute, Carnegie Mellon University. Image Understanding Research in the TVL is supported in part by the Defense Advanced Research Project Agency, DOD, through ARPA order No. 4976, and monitored by the Air Force Avionics Laboratory under contract F33615-87-C-1499.

Yoshimasa Fujiwara was on leave of absence from Matsushita Electric Works, Ltd., Osaka Japan.

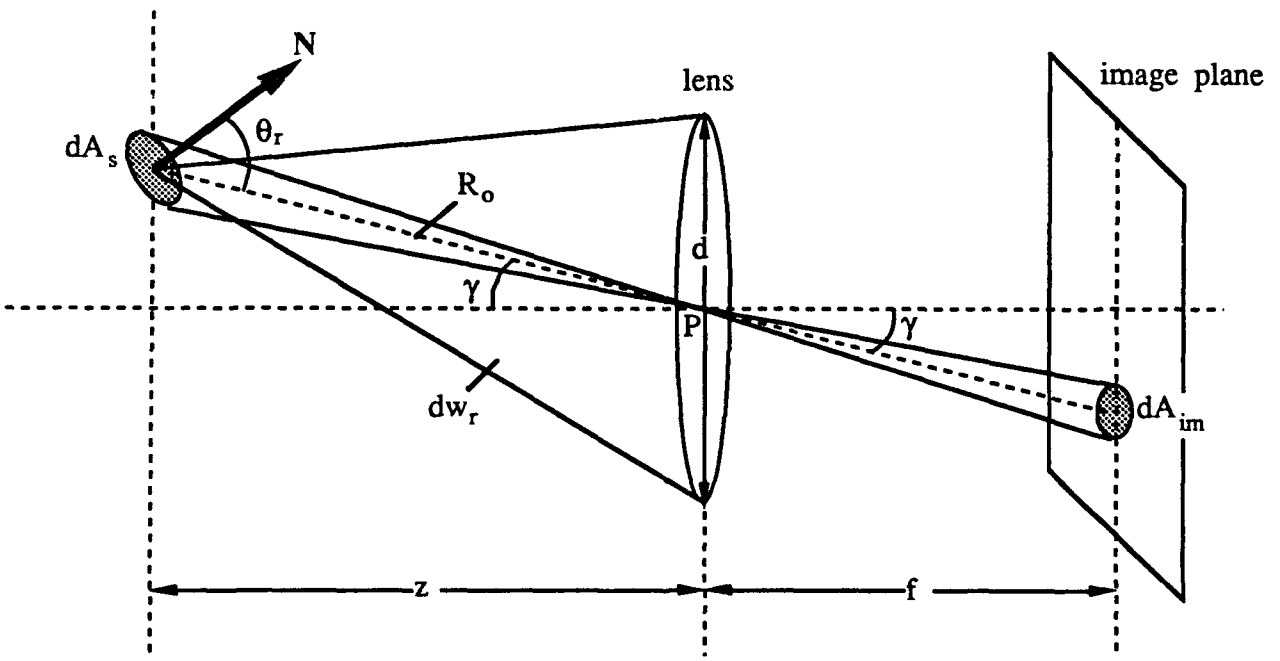


Figure 1: Image formation: light waves radiated by the surface area dA_s and gathered by the lens are projected onto an area dA_{im} on the image plane. Adapted from [5].

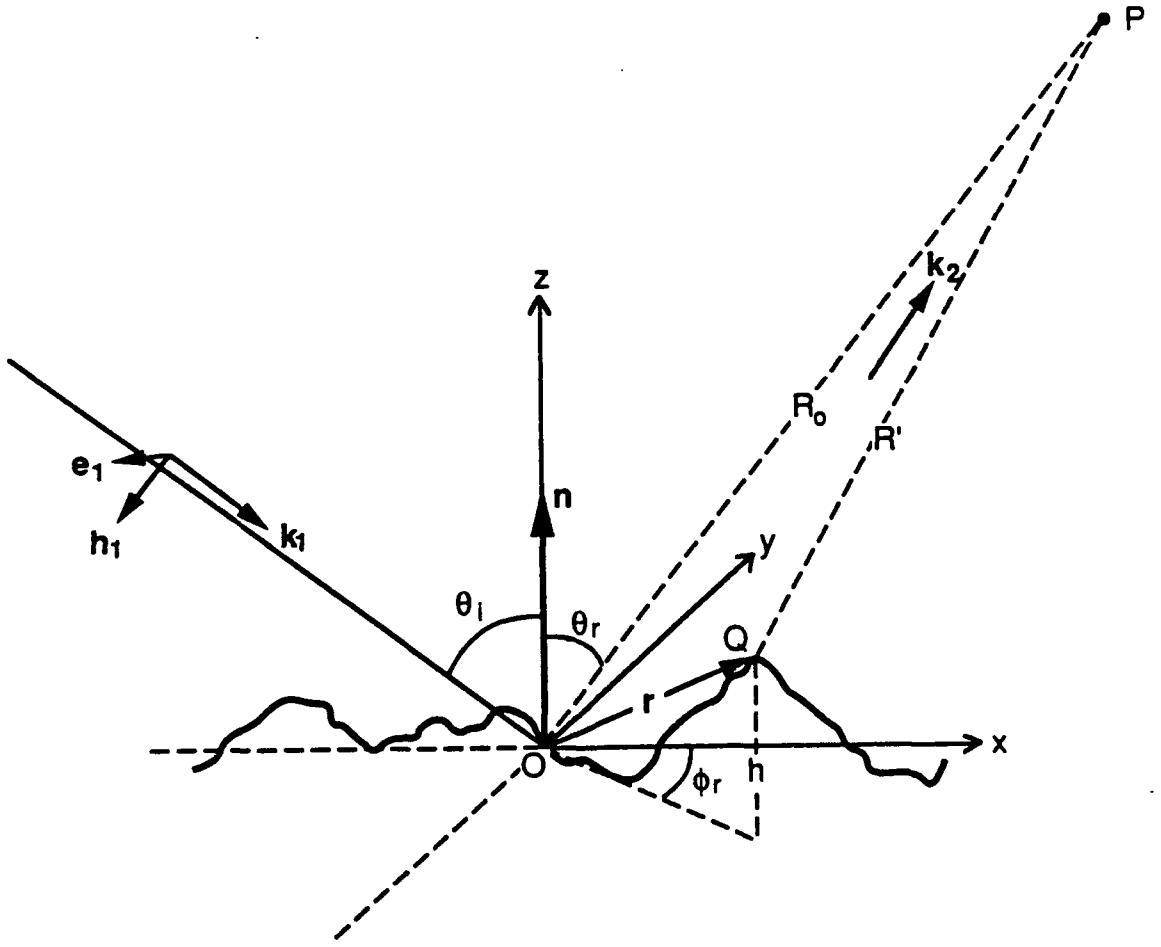


Figure 2: Coordinate system to derive the Beckmann-Spizzichino model

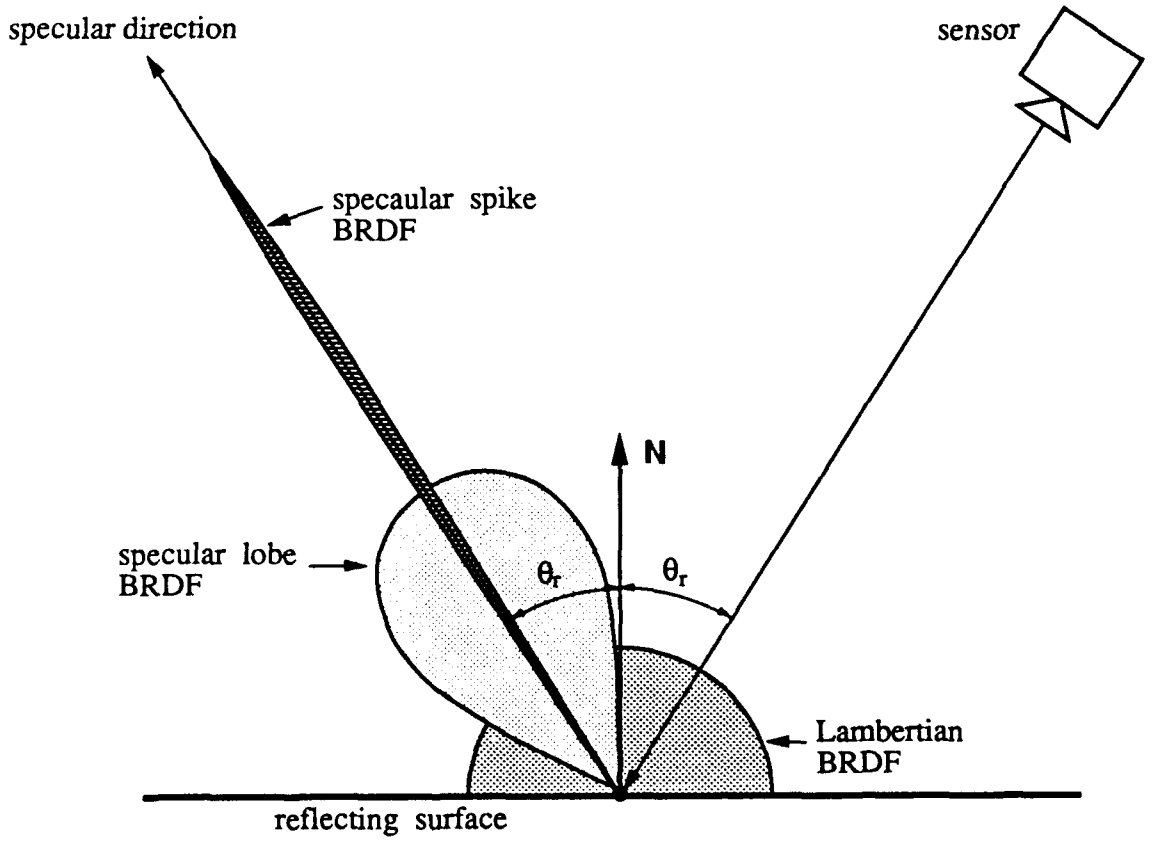


Figure 3: Polar plots of the three reflection components as function of the source angle for a fixed viewing direction.

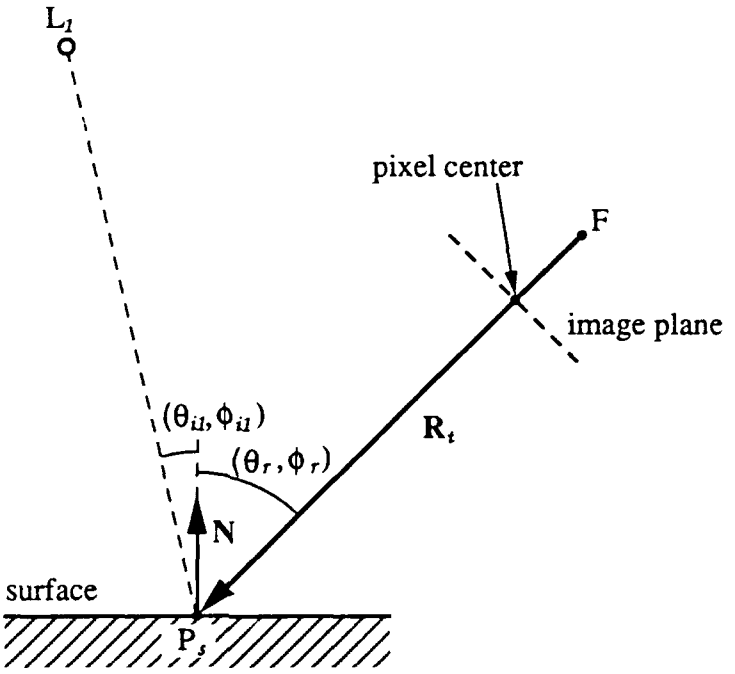


Figure 4: Ray tracing geometry (1 light, no another surface)

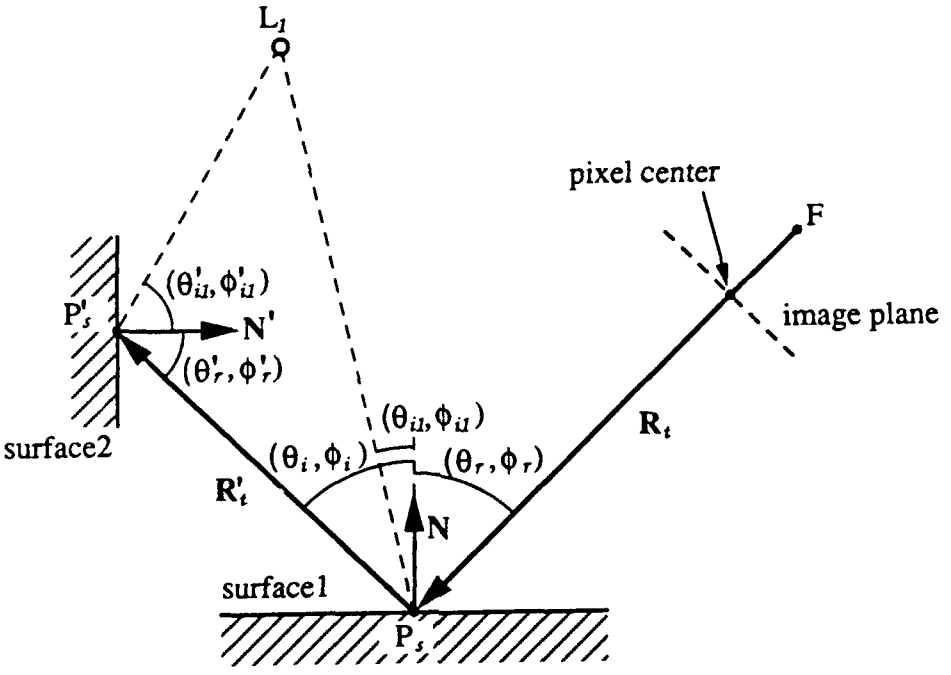


Figure 5: Ray tracing geometry (1 light, one more surface)

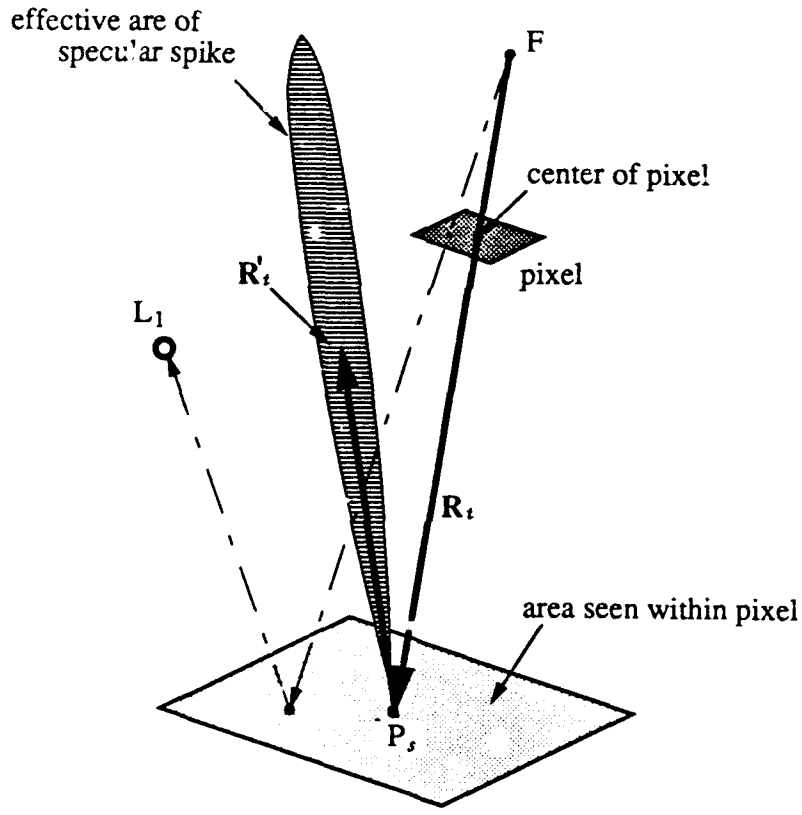


Figure 6: Missing of light source in specular spike case

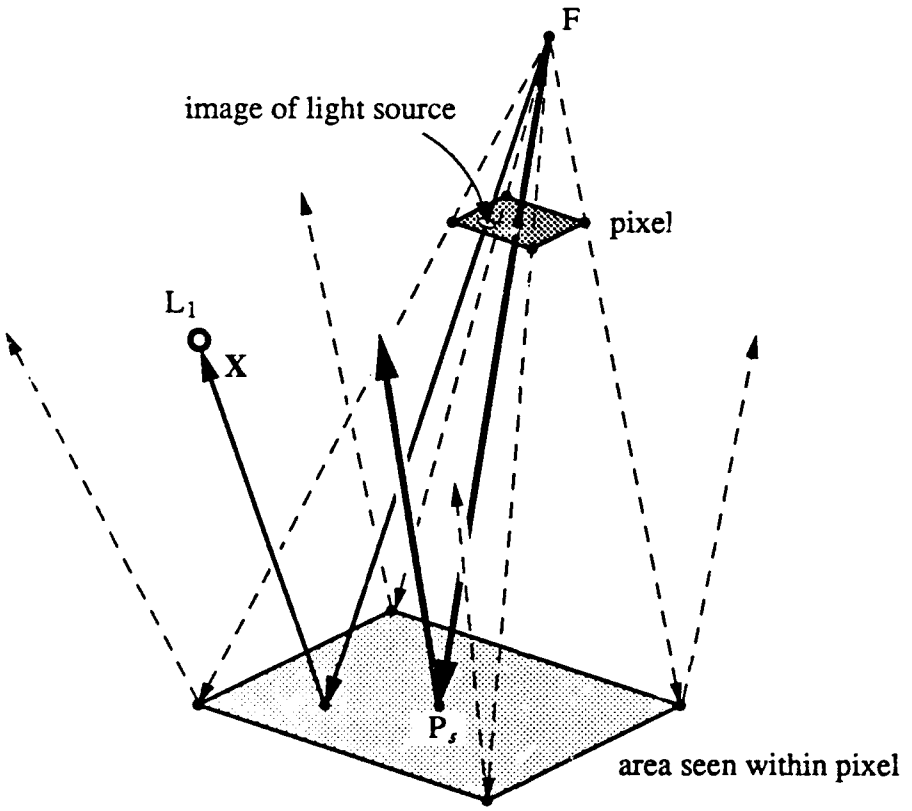


Figure 7: Recovery of light source in specular spike case

Figure 8: Ray tracing image: intermediate roughness surface

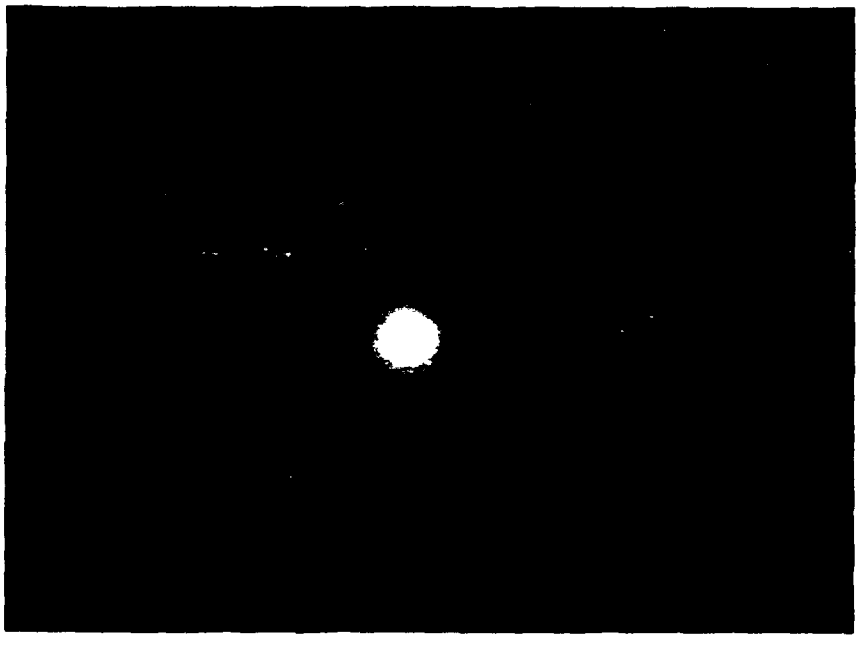


Figure 9: Actual image: intermediate roughness surface

Figure 10: Ray tracing image: smooth surface

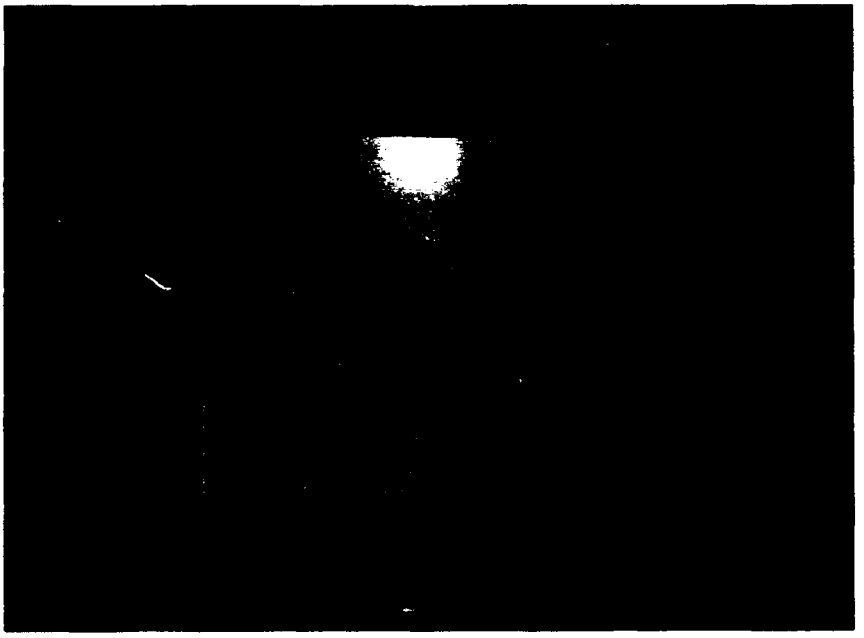


Figure 11: Ray tracing image: intermediately rough ball and plane

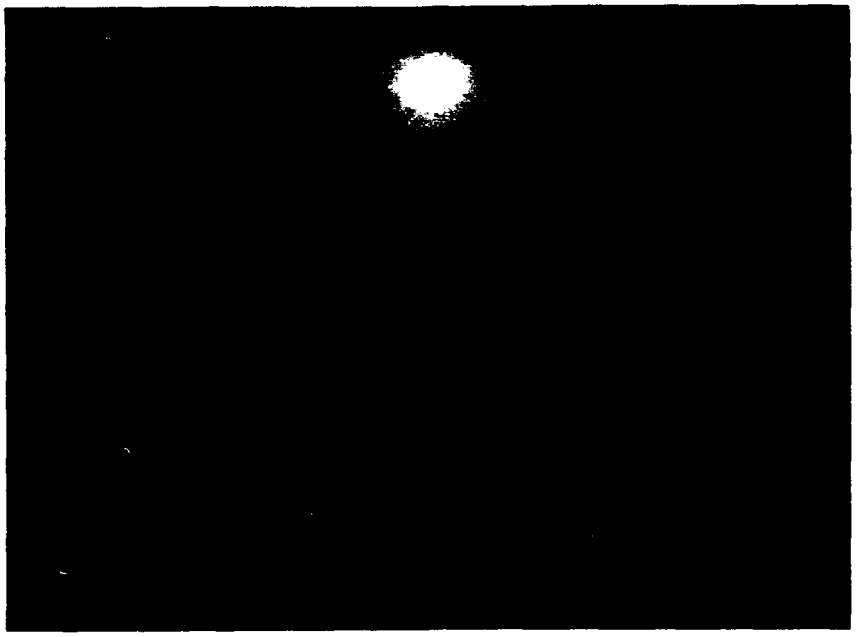


Figure 12: Intermediately rough ball and plane

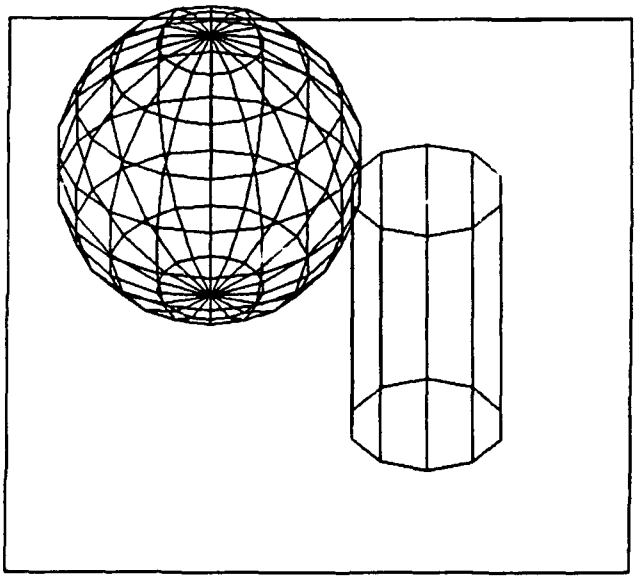


Figure 13: Line drawing of a scene generated by Vantage

Figure 14: Appearance generated by the simulator

Figure 15: Depth map generated by the simulator

Figure 16: Distribution of primal face ID number

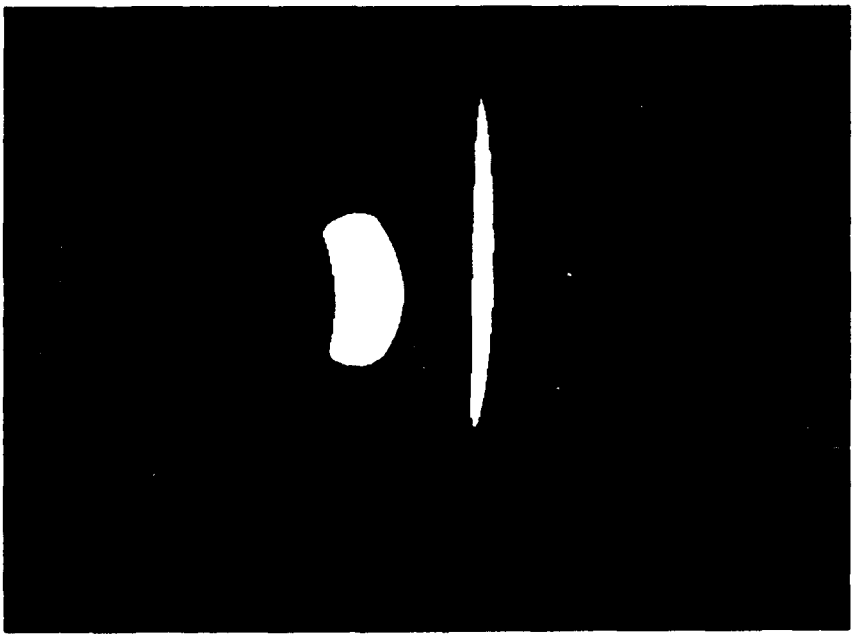


Figure 17: Distribution of secondary face ID number

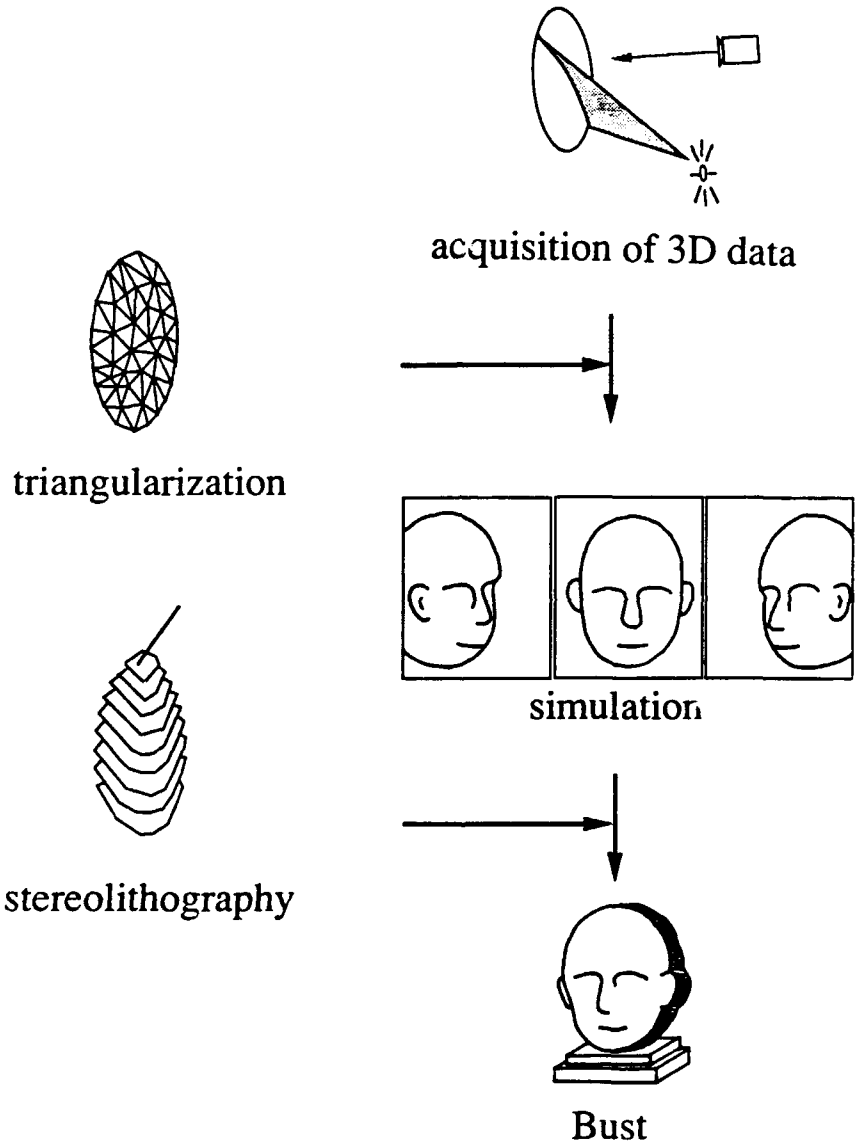


Figure 18: Bust generation system

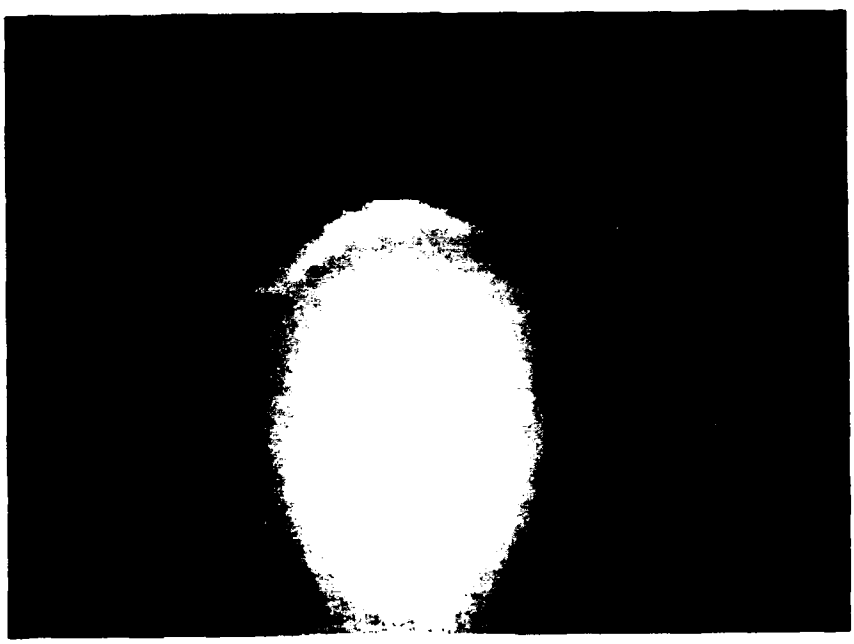


Figure 19: Input Images; (a) brightness image; (b) isoplot image of a depth map

Figure 20: Simulated images; (a) plaster image; (b) stainless steel image

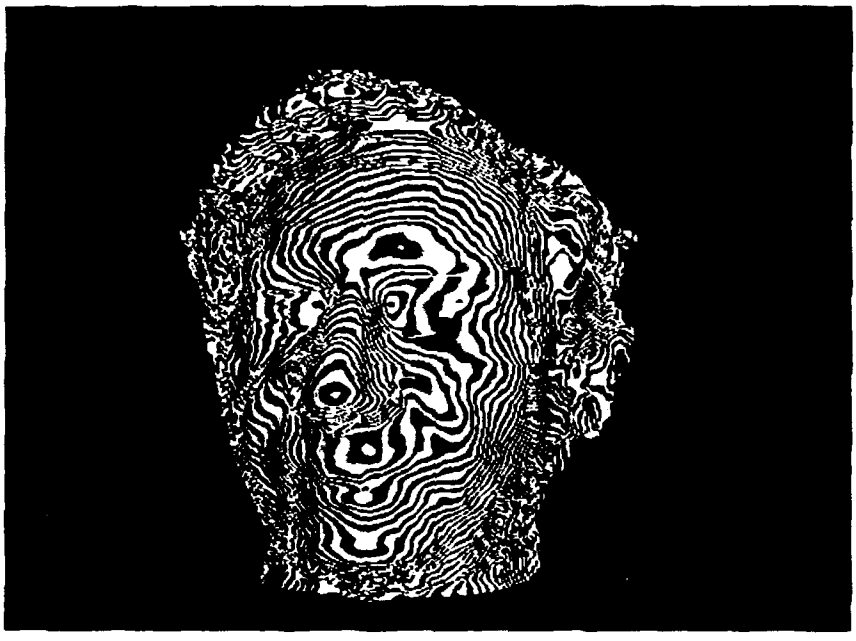


Figure 21: Distance distribution